Week 15, Lecture 30

István Albert

Biochemistry and Molecular Biology and Bioinformatics Consulting Center

Penn State
Discussion on the project

• Some observations on the project

• What went well and what went wrong

• Assembly is not well defined - small changes in hash sizes → different results
Assembly on forward strand
Assembly on reverse strand
git clone https://github.com/lh3/bioawk.git
www.biopieces.org

The Biopieces are a collection of bioinformatics tools that can be pieced together in a very easy and flexible manner to perform both simple and complex tasks. The Biopieces work on a data stream in such a way that the data stream can be passed through several different Biopieces, each performing one specific task: modifying or adding records to the data stream, creating plots, or uploading data to databases and web services. The Biopieces are executed in a command line environment where the data stream is initialized by specific Biopieces which read data from files, databases, or web services, and output records to the data stream that is passed to downstream Biopieces until the data stream is terminated at the end of the analysis as outlined below.

read_data | calculate_something | write_results

The following example demonstrates how a next generation sequencing experiment can be analyzed – including removal of adaptor sequence, determining the number of unique sequences, mapping to a specified genome, and uploading the data to the UCSC genome browser for further analysis:

read_fastq -i data.fq | remove_adaptor -a TCGTATGCCAGTC -m 2 | grab -e ADAPTORS.POS > -1 | count_vals -k SEQ | uniq_vals -k SEQ_NAME,SEQ_COUNT | merge_vals -k SEQ_NAME | vmatch_seq -g hg18 | upload_to_ucsc -d hg18 -t my_data -x # Initialize data stream from a file. # Remove adaptor sequence allowing for 2 mismatches. # Get all entries where an adaptor sequence was found. # Determine the occurrences of all sequences. # Get all entries with a unique sequence. # Append the sequence count to the sequence name. # Map the sequences to the Human genome using vmatch. # Upload the mapping results to the UCSC Genome Browser.

The advantage of the Biopieces is that a user can easily solve simple and complex tasks without having any programming experience. Moreover, since the data format used to pass data between Biopieces is text based, different developers can quickly create new Biopieces in their favorite programming language – and all the Biopieces will maintain compatibility.

There are currently ~160 Biopieces.
- To learn more about Biopieces have a look at the Biopieces Introduction.
- To browse the available Biopieces go to the Biopieces Wiki.
- If you want to install the Biopieces go to the Biopieces Installation Instructions.
- If you want to contribute Biopieces go to the Biopieces HowTo.
- Browse publications using Biopieces here.

For important messages, questions, discussion, and suggestions join the Biopieces Google Group.
Introducing Savant

The Savant Genome Browser is a desktop visualization tool for genomic data. It was primarily developed for visualizing high throughput (aka next generation) sequencing data, although it can be used to visualize virtually any genome-based sequence, point, interval, or continuous dataset.

Questions, comments, or suggestions are welcome via e-mail to: support [at] savantbrowser [dot] com.

Citation

Savant. Genome Browser for High Throughput Sequencing Data
Marc Fiume; Vanessa Williams; Andrew Brook; Michael Biudno
http://compbio.cs.toronto.edu/savant/
Grouping mutations/Gene=f(sample)

GroupByGene is a small C++ tool grouping the data:
- CHROM
- POS
- REF
- GENE
- SAMPLE

by gene=f(sample). This tool is available on github: https://github.com/lindenb/ccansbox/blob/master/src/groupbygene.cpp.

Example:

```
# cat input.txt

chr1 19 A T gene1 15d11
chr1 19 A T gene1 15d12
chr2 160 C G gene2 1id13
chr3 216 A T gene3 1id11
chr3 211 C G gene3 1id12
chr4 590 C A gene4 1id13
```

Calling `groupbygene`: 
Dalliance :: Home

Dalliance is an interactive genome viewer which runs directly in your web browser. If you are running a modern browser with Javascript enabled, you should see it running on this page.

![Genome Browser Screenshot](image)

The aim is for a high level of interactivity. Drag to scroll ("Google Maps" style), or try a panning gesture if your input device supports this. You can add more data from the DAS registry by clicking the "Add track" button in the toolbar, or follow the links in the sidebar to browse other genomes or more data.

Dalliance requires a modern web browser with good support for Scalable Vector Graphics. Firefox 3.6+, Google Chrome, and Safari 5 are all tested regularly and seem to work nicely. Internet Explorer won't work at all, although we're optimistic that we'll be able to support IE9 (but not earlier versions) in the future.

Dalliance also doesn't currently support touch based hardware such as the iphone, ipod and ipad. We've looked at adding support but currently there are serious browser performance issues with the webkit which make scrolling ineffective.

filo - Useful FILE and stream Operations

The following tools are available as part of the filo package. More to come...

Contents

- filo - Useful FILE and stream Operations
  - groupBy
  - shuffle
  - stats

groupBy

groupBy is a useful tool that mimics the "groupBy" clause in database systems. Given a file or stream that is sorted by the appropriate "grouping columns", groupBy will compute summary statistics on another column in the file or stream. This will work with output from all BEDTools as well as any other tab-delimited file or stream.

You specify a list of columns that should be "grouped" with the -g parameter (e.g., -g 2,3,4 will group on the second through fourth columns). You then specify column(s) that should be summarized or "operated upon" for each group with the -c parameter (e.g., -c 2 or -c 2,3 or -c 2,2,2,5). Finally, you specify what operations should be applied to the list of columns in -c.

Here is the current list of the available operations:

1. sum - numeric only
2. count - numeric or text
3. min - numeric only
4. max - numeric only
5. mean - numeric only
6. stddev - numeric only
7. median - numeric only
8. mode - numeric or text
9. antmode - numeric or text collapse (i.e., print a comma separated list) - numeric or text
10. freqasc - print a comma separated list of values observed and the number of times they were observed. Reported in ascending order of frequency.
11. freqdesc - print a comma separated list of values observed and the number of times they were observed. Reported in descending order of frequency.
12. collapse - print a comma separated list of each value in the grouped column.
13. concat - concatenate each value in the grouped column into a single string.

And here are some usage examples. I hope you find this utility to be of use in your work. I have found it to be a huge time saver.
Stampy

Investigators: Gerton Lunten and Martin Goodson

Description: Stampy is a package for the mapping of short reads from illumina sequencing machines onto a reference genome. It’s recommended for most workflows, including those for genomic resequencing, RNA-Seq and Chip-seq. Stampy excels in the mapping of reads containing that contain sequence variation relative to the reference, in particular for those containing insertions or deletions. It can map reads from a highly divergent species to a reference genome for instance. Stampy achieves high sensitivity and speed by using a fast hashing algorithm and a detailed statistical model. Stampy has the following features:

- Maps single, paired-end and mate pair Illumina reads to a reference genome
- Fast: about 20 Gbase per hour in hybrid mode (using BWA)
- Low memory footprint: 2.7 Gb shared memory for a 3Gbase genome
- High sensitivity for indels and divergent reads, up to 10-15%
- Low mapping bias for reads with SNPs
- Well calibrated mapping quality scores
- Input: Fastq and Fasta; gzipped or plain
- Output: SAM, Maq’s map file
- Optionally calculates per-base alignment posteriors
- Optionally processes part of the input
- Handles reads of up to 4500 bases
Tablet - Next Generation Sequence Assembly Visualization

Tablet is a lightweight, high-performance graphical viewer for next generation sequence assemblies and alignments.

Visit the download page to get the latest version of Tablet (1.11.08.10; 10th August 2011) now.

Using UCSC Genome Browser Track Hubs

Table of Contents:

- What Are Track Hubs?
- Viewing Track Hubs in the Browser
- Setting Up Your Own Track Hub
- Debugging and Updating Track Hubs
- Registering a Track Hub with UCSC

Search the Genome Browser help pages: [ ] Submit

Questions and feedback are welcome.

What Are Track Hubs?

Track hubs are web-accessible directories of genomic data that can be viewed on the UCSC Genome Browser alongside native annotation tracks. Hubs are a useful tool for visualizing a large number of genome-wide data sets. For example, a project that has produced several wiggle plots of data can use the hub utility to organize the tracks into composite and super-tracks, making it possible to show the data for a large collection of tissues and experimental conditions in a visually elegant way, similar to how the ENCODE native data tracks are displayed in the browser.

The track hub utility allows efficient access to data sets from around the world through the familiar Genome Browser interface. Browser users can display tracks from any public track hub that has been registered with UCSC. Additionally, users can import data from unlisted hubs or can set up, display, and share their own track hubs.

The data underlying the tracks in a hub reside on the remote server of the data provider rather than at UCSC. The data are stored in compressed binary indexed files in bigBed, bigWig, or BAM format that contain the data at several resolutions. When a hub track is displayed in the Genome Browser, only the relevant data needed to support the view of the current genomic region are transmitted rather than the entire file.
Welcome to the MochiView website!

What is MochiView?

MochiView is Java software that integrates browsing of genomic sequences, features, and data with DNA motif visualization and analysis. The FEATURE OVERVIEW page provides a detailed listing of MochiView’s features. A manuscript describing MochiView has also been published:

MochiView: versatile software for genome browsing and DNA motif analysis

How do I get started?

You can get up and running in just two steps. Listed before each step is a link to the relevant section of the website (also found on the left side of each page). You can return to this page at any time by clicking the GETTING STARTED link.

SYSTEM REQUIREMENTS: Check this link to make sure that your computer can run MochiView. Most importantly, follow the instructions to make sure that your Java version is recent enough to run the software.

SOFTWARE & TUTORIAL DOWNLOAD: This section provides the necessary links to download the software and manual. I strongly recommend that new users download the tutorial database/instructions and use the tutorial to become familiar with the software.

What are those other links?

Here is an overview of the contents of the other sections of the website.

NEWS: Contains information about software and website content updates.
Scythe - A very simple adapter trimmer (version 0.98 BETA)

Contact: Vince Buffalo vsbuffaloAAAA@gmail.com (with the poly-A tail removed)

Copyright (c) 2011 The Regents of University of California, Davis Campus.

About

Scythe uses a Naive Bayesian approach to classify contaminant substrings in sequence reads. It considers quality information, which can make it robust in picking out 3-end adapters, which often include poor quality bases.

Most next generation sequencing reads have deteriorating quality towards the 3-end. It's common for a quality-based trimmer to be employed before mapping, assemblies, and analysis to remove these poor quality bases. However, quality-based trimming could remove bases that are helpful in identifying (and removing) 3-end adapter contaminants. Thus, it is recommended you run Scythe before quality-based trimming, as part of a read quality control pipeline.

The Bayesian approach Scythe uses compares two likelihood models: the probability of seeing the matches in a sequence given contamination, and not given contamination. Given that the read is contaminated, the probability of seeing a certain number of matches and mismatches is a function of the quality of the sequence. Given the read is not contaminated (and is thus assumed to be random sequence), the probability of seeing a certain number of matches and mismatches is chance. The posterior is calculated across both these likelihood models, and the class (contaminated or not contaminated) with the maximum posterior probability is the class selected.

Requirements

Scythe can be compiled using GCC or Clang; compilation during development used the latter. Scythe relies on Heng Li's kseq.h, which is bundled with the source.

Scythe requires Zlib, which can be obtained at http://www.zlib.net/.

Building and Installing Scythe

To build Scythe, enter:

```bash
make build
```

Then, copy or move "scythe" to a directory in your $PATH.

Usage
How to Install and Use Miropeats

***updated August 13, 2010***

Contents

- Introduction - What the program can do.
- Installation
- Typical Usage - How to type the commands.
- Example Output - What the program can produce.
- Key to Graphics - How to understand the pictures.
- Bugs
- Disclaimer
- References
- Contact Information

Introduction

Miropeats discovers regions of sequence similarity amongst any set of DNA sequences and then presents this similarity information graphically. Sequence similarity searching is a very general tool that forms the basis of many different biological sequence analyses but it is limited by the verbosity of traditional alignment presentation styles. Miropeats enhances the utility of conventional DNA sequence comparisons when looking at long lengths of sequence similarity by summarizing extensive large scale sequence similarities on a single page of graphics.

The descriptive abilities of Miropeats open research opportunities that would not be possible, or would be tedious, or difficult to do otherwise. Examples include comparing the repeat structures of entire chromosomes, visualising overlapping sequence fragments in a contig assembly project and comparing the products of different contig assembly programs. Miropeats was originally written to help contig assembly projects at the Genome Sequencing Center in St. Louis, Missouri, USA, where it was found to be useful for many different roles. The intrinsic inerstability of a string of 40,000 characters picked from an alphabet of only 4 letters (a typical cosmid assembly project) is made worse because the shotgun sequencing strategy starts with the original contiguous 40Kb DNA sequence split into an 800 piece puzzle. Miropeats helps shotgun assembly, not by solving the puzzle itself, but by helping the researcher gain an overall understanding of the task presented to them. Miropeats can do this because it draws a simple graphic that shows potential joins, cosmid overlaps, and also distinguishes tandem repeats, inverted repeats, oligo repeats and palindromes from each other.

Miropeats has options to look at all repeated DNA sequence segments (Default) or one can choose to see only those repeated sequences with either both copies on a single sequence, or both copies on different sequences. The program also has an adjustable threshold that lets the user choose what length of DNA sequence similarity should be considered significant and worth displaying. This facility allows Miropeats to be used for analyzing different features in sequences ranging from...
EMBOSS is funded from May 2009 by BBSRC grant BBR/G02264X/1

Funded from May 2006 to April 2009 by BBSRC grant BB/D018358/1

About EMBASSY • EMBoss • Groups • Proposed

Hundreds of useful, well documented applications for molecular sequence and other analyses... more >

GUIs Jemboss • GUIs • Web • Others

We support the Jemboss GUI but many others are available... more >

Servers Portals • Servers • Mirrors • Misc

Many EMBoss portals, servers and mirrors are available... more >

Downloads Stable release • Developers (CVS) version • Getting started

EMBOSS is open source software and is freely available to all... more >

Licence Licensing terms
Readseq: Read & reformat biosequences .. web service

Readseq was written originally around 1989 a component of a sequence analysis program, but when I added a small, simple command-line interface, it took on a life of its own as a conversion program for bioinformatics. It's main contribution to bioinformatics is it takes on the job of guessing what your input biosequence data format is, and converting it to what your software knows how to handle. See here readseq-help.html and Readseq2-help.html for extended help.

What you need to use it is

1. readseq.jar the java archive of this program. (don't unpack this jar on your computer).
2. a Java runtime system on your computer, versions 1.1,1.2,1.3,1.4,1.5.
3. for OLD MacOS 9+, take also ReadseqApp, a small Mac application to run readseq without a command line (B>do un-binhex and un-stuffit this one).

This version includes a Graphic User Interface (GUI) for those who prefer not to learn the many command line options, or who's workstation lacks a command-line interface (which probably includes most but the hardy bioinformaticians :). Double-click the readseq.jar file to launch or Run as

java -jar readseq.jar

If you do like command line interfaces, see the docs, the brief doc is

java -cp readseq.jar help
java -cp readseq.jar run -- for command-line
java -cp readseq.jar app -- for GUI application
USEARCH

About USEARCH
USEARCH is a unique high-throughput sequence analysis tool. It is a distributed as single binary program that implements a suite of algorithms comparable to BLASTN, BLASTP, BLASTX, BLASTCLUST, CD-HIT, CD-HIT-EST, CD-HIT-2D, CD-HIT-EST-2D, CD-HIT-OTU, CD-HIT-454, ChimeraSlayer, Perseus, RAPSearch and more. It supports a rich set of sequence matching options, including E-values, identity coverage (fraction of query or target sequence covered by the alignment) and maximum gap length, and a range of output file formats including FASTA, BLAST-like, user-defined tabbed text and a native format designed for clustering applications. Supported alignment styles include local (gapped and ungapped), like BLAST, and global, which is most often used in clustering applications. User-settable parameters allow tuning of substitution scores, gap penalties and Karlin-Atchul statistics.

USEARCH is under rapid development; new features are added every few weeks. To stay informed, sign up for the mailing list.

Algorithms
See benchmark test results for performance comparisons.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database search</strong></td>
<td>Fast, heuristic local search (like BLAST). Fast, heuristic global search.</td>
<td>Supports fast &quot;top-hit&quot; and &quot;top-few hits&quot; modes and search of entire</td>
</tr>
<tr>
<td></td>
<td>Smith-Waterman local search (like SSEARCH). Needleman-Wunsch global search</td>
<td>database. Top-hit modes achieve search speeds up that can be orders</td>
</tr>
<tr>
<td></td>
<td>(like NEEDLE).</td>
<td>of magnitude faster than conventional methods like BLAST.</td>
</tr>
<tr>
<td><strong>Clustering</strong></td>
<td>Greedy representative sequence clustering (like CD-HIT), and / or centroid</td>
<td>Supports any sort order, e.g. length sort for reducing redundancy, or</td>
</tr>
<tr>
<td></td>
<td>sequence construction (by consensus).</td>
<td>quality / abundance sort for error correction / denoising. CD-HIT</td>
</tr>
<tr>
<td><strong>Search+clustering</strong></td>
<td>Single-pass method that (i) assigns matching sequences to a database and</td>
<td>supports length sort only.</td>
</tr>
<tr>
<td></td>
<td>(ii) clusters sequences that don’t match.</td>
<td></td>
</tr>
</tbody>
</table>

Typically used in OTU applications.
Bambino: a variant detector and alignment viewer for next-generation sequencing data in the SAM/BAM format

Table of contents

- Introduction
- Launching bambino
- Example data / walkthrough
- Viewer display overview
- Navigation
- Context menu
- Variant detection
- Command-line usage / utilities
- Citation
- Troubleshooting / submitting test data
- Change log

Introduction

"Bambino" is a graphical viewer and variant detector for next-generation sequencing files in SAM/BAM format. Its primary features:

- detects SNPs, insertions, and deletions directly from BAM files using a configurable process. This may be run from the command line (see BAM utilities documentation) or interactively from within the viewer.
- can dynamically pool data from multiple BAM files: view and analyze data from tumor/normal pairs, multiple runs, or even platforms. The variant detector takes advantage of pooling as well, producing output suitable for somatic mutation detection.
- Reconciles all indels in the dataset into a unified, padded alignment with the reference sequence, padding the reads and the reference as necessary. While the SAM/BAM specification allows for padded alignments, most BAM files contain unpadded read-mapping data. Bambino essentially converts these unpadded alignments into padded ones. This postprocessing provides a complete and unified view of the assembly, even of reads having different indel sizes at the same location. This is useful for visualizing features such as STRPs, and debugging the alignments produced by read-mapping programs.
- displays mappings of dbSNP SNP sites and reference protein sequences, using UCSC genome annotations. The database connection is configurable; it can load
LASTZ  Release 1.02.00, built January 12, 2010

TABLE OF CONTENTS

- Introduction
- Availability
- Installation
- Overview of Processing Stages and Terminology
- Examples
  - Comparing a Human Chromosome and a Chicken Chromosome
  - Aligning Shotgun Reads to a Human Chromosome
  - Seeds, HSPs, Gapped Alignments, Chaining
  - Aligning a Sequence With Itself
- Command-line Syntax
  - Where to Look
  - Scoring
  - Indexing
  - Seeding
  - HSPs (Gap-free Extension)
  - Chaining
  - Gapped Extension
  - Back-end Filtering
  - Interpolation
  - Output
  - Housekeeping
  - Shortcuts for Yasra
  - Help
  - Sequence Specifiers
- Processing Stages in Detail
  - Target Sequence Input
mrFAST
Micro Read Fast Alignment Search Tool

mrFAST and mrFAST are designed to map short reads generated with the Illumina platform to reference genome assemblies, in a fast and memory-efficient manner.

- **mrFAST**: micro-read Fast Alignment Search Tool.
  - Currently Supported Features:
    - Output in SAM format
    - Indels up to 4bp
    - Paired-end mapping
      - Discordant option to generate mapping file ready for VariationHunter to detect structural variants.
    - One end anchored (OEA) map locations for novel sequence insertion detection with NovelSeq
  - Upcoming Features:
    - Automatic detection of FASTQ quality offset. Current default is 33.
    - Matepair library mapping (long inserts with RF orientation).

2011-09-12: mrFAST version 2.1.0.0 release:

- Several bugs are fixed.
- New parameter –maxoa now limits the maximum number of locations for one end anchored (OEA) sequences. This will help substantially reduce the output file size. For NovelSeq use, we recommend to set this parameter to at least 100.
- New parameter –maxoa now limits the maximum number of locations for discordant read pairs in the DIVET file. This will help substantially reduce the output file size. For VariationHunter use, we recommend to set this parameter to at least 300.

2010-09-28: mrFAST version 2.0.0.5 release:

- Several minor bugs are fixed.
- Sequence and quality information of the unmapped ends of OEA pairs are added to the SAM-compatible OEA output file for NovelSeq. Next release of NovelSeq will be able to parse this file. Please contact NovelSeq authors if you need help with format conversion in the mean time.
- User manual uploaded.

2010-08-28: mrFAST version 2.0.0.3 release:

- One end anchored (OEA) map locations for novel sequence insertion detection with NovelSeq added. The output format is SAM, please contact NovelSeq authors if you need help with format conversion.

2010-07-24: mrFAST version 2.0.0.2 release:
HTSeq: Analysing high-throughput sequencing data with Python

- Overview
  - Documentation overview
  - Author
  - License
- Prerequisites and installation
  - Prerequisites
  - Download
  - Installing a binary package
  - Installing a source package
- A tour through HTSeq
  - Reading in reads
  - Genomic intervals and genomic arrays
  - Calculating coverage vectors
  - Counting reads by genes
  - Mapping structural variants to genes
  - And much more
- A detailed use case: TSS plots
  - Using the full coverage
  - Using indexed BAM files
  - Streaming through all reads
- Sequences and FASTA/FASTQ files
  - Sequence
  - SequenceWithQualities
  - FastaReader
NGS mapper ROC curves

The ROC curves

Simulated data

100,000 reads (read pairs) are simulated the human genome with *wgsim*. In this simulation, we first simulate a diploid genome containing about 28.6 million SNPs and short INDELS and then simulate error free reads from the diploid genome. Although reads are error free, many reads cannot be perfectly mapped to the reference genome due to the presence of variations. The exact *wgsim* command lines are (the first for single-end and the second for paired-end):

1. wgsim -N 100000 -r 0.01 -1 100 -511 -00 -00 hs37d4.fa r1.fq /dev/null
2. wgsim -N 1000000 -r 0.01 -1 100 -2 100 -511 -00 hs37d4.fa r1.fq r2.fq

Evaluated programs and command lines

Default configurations are attempted unless the default certainly fails or the documentation explicitly suggests better